Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Please amend the claims as follows:

1. (Currently Amended) An integrated tracing and logging system employed within a system of computers interconnected through a network comprising:

a tracing module associated with specified program code regions of an application, the tracing module to receive and process tracing method calls generated by the application when the specified program code regions are executed, the tracing module being an instance of a first sub-class of a class; and

a logging module associated with specified categories of the system, the logging module to receive and process logging method calls from components associated with the categories, the logging module being an instance of a second sub-class of the class, the first sub-class being different than the second sub-class;

a trace filter to filter the tracing method calls according to a specified trace message filtering policy wherein the specified trace message filtering policy comprises blocking a trace message from a first output destination but allowing the trace message to be sent to a second output destination.

2. (Original) The system as in claim 1 wherein the tracing module further comprises trace severity logic to identify a tracing severity level associated with the tracing method calls and to process the tracing method calls based on the tracing severity level.

3. (Original) The system as in claim 2 wherein the trace severity logic processes the tracing method calls by comparing the tracing severity level of the method calls to a tracing severity threshold, wherein if the tracing severity level

crosses the tracing severity threshold, a trace message is sent to a first output destination.

4. (Original) The system as in claim 2 wherein the logging module further comprises log severity logic to identify a logging severity level associated with the logging method calls and to process the logging method calls based on the logging severity level.

5. (Original) The system as in claim 4 wherein the log severity logic processes the logging method calls by comparing the logging severity level of the method calls to a logging severity threshold, wherein if the logging severity level crosses the logging severity threshold, a log message is sent to a second output destination.

6. (Original) The system as in claim 5 wherein the first and/or second output destination comprises a display console.

7. (Original) The system as in claim 5 wherein the first and/or second output destination comprises a trace and/or log file, respectively.

8. (Original) The system as in claim 5 wherein the first output destination is equivalent to the second output destination.

9. (canceled).

10. (canceled).

11. (canceled).

12. (Original) The system as in claim 5 further comprising:

a log filter to filter the logging method calls according to a specified log filtering policy.

13. (Currently Amended) The system as in claim 12 wherein filtering comprises precluding the [[trace]] log message from being sent to the first output destination.

14. (Original) The system as in claim 12 wherein the specified log filtering policy comprises blocking the log message from the first output destination but allowing the log message to be sent to a second output destination.

15. (Original) The system as in claim 1 further comprising a plurality of trace message and/or log message formatters to convert the trace method calls and/or the log method calls to a specified one or more trace and/or log message formats, respectively.

16. (Original) The system as in claim 15 wherein at least one of the trace and/or log message formats comprises a human-readable format for displaying on a display console.

17. (Previously Presented) The system as in claim 1 wherein the class is a controller class, the first sub-class is a tracing subclass to the controller class and the second sub-class is a logging subclass to the controller class.

18. (Currently Amended) An integrated logging and tracing system employed within a system of computers interconnected through a network comprising:

integrated logging and tracing means, the tracing means associated with specified program code regions of an application, the tracing means to receive and

process tracing method calls generated by the application when the specified program code regions are executed; and

the logging means associated with specified categories of the system, the logging means to receive and process logging method calls from components associated with the categories, the means and tracing means being respective instances of different sub-classes of the same class;

filtering means to filter the tracing method calls and the logging method calls according to a specified trace message filtering policy and logging message filtering policy, respectively, the filtering means preventing the calls from reaching respective trace files and log files.

19. (Original) The system as in claim 18 wherein the tracing means further comprises trace severity logic to identify a tracing severity level associated with the tracing method calls and to process the tracing method calls based on the tracing severity level.

20. (Original) The system as in claim 19 wherein the trace severity logic processes the tracing method calls by comparing the tracing severity level of the method calls to a tracing severity threshold, wherein if the tracing severity level crosses the tracing severity threshold, a trace message is sent to a first output destination.

21. (Original) The system as in claim 19 wherein the logging means further comprises log severity logic to identify a logging severity level associated with the logging method calls and to process the logging method calls based on the logging severity level.

22. (Original) The system as in claim 21 wherein the log severity logic processes the logging method calls by comparing the logging severity level of the

method calls to a logging severity threshold, wherein if the logging severity level is equal to or greater than the logging severity threshold, a log message is sent to a second output destination.

23. (canceled).

24. (Original) The system as in claim 23 further comprising formatting means to convert the trace method calls and/or the log method calls to a specified one or more trace and/or log message formats, respectively.

25. (Original) The system as in claim 24 wherein at least one of the trace and/or log message formats comprises a human-readable format for displaying on a display console.

26. (Currently Amended) A method employed within a system that includes a network comprising:

defining a class hierarchy comprising a controller class, a tracing sub-class, and a logging sub-class, wherein the controller class is a parent class to the tracing sub-class and the logging sub-class;

creating an instance of the tracing sub-class associated with specified program code regions of an application, the tracing instance to receive and process tracing method calls generated by the application when the specified program code regions are executed; and

creating an instance of the logging sub-class associated with specified categories of the system, the logging instance to receive and process logging method calls from components associated with the categories;

filtering the tracing method calls and the logging method calls according to a specified trace message filtering policy and logging message filtering policy,

respectively, the filtering preventing the calls from reaching respective trace files and log files.

27. (Original) The method as in claim 26 wherein the instance of the tracing sub-class further identifies a tracing severity level associated with the tracing method calls and to process the tracing method calls based on the tracing severity level.

28. (Original) The method as in claim 27 wherein the instance of the tracing subclass processes the tracing method calls by comparing the tracing severity level of the method calls to a tracing severity threshold, wherein if the tracing severity level is equal to or greater than the tracing severity threshold, a trace message is sent to a first output destination.

29. (Original) The method as in claim 27 wherein the instance of the logging sub-class identifies a logging severity level associated with the logging method calls and to process the logging method calls based on the logging severity level.

30. (Original) The method as in claim 29 wherein the instance of the logging subclass processes the logging method calls by comparing the logging severity level of the method calls to a logging severity threshold, wherein if the logging severity level is equal to or greater than the logging severity threshold, a log message is sent to a second output destination.

31. (canceled).

32. (Original) The method as in claim 31 further comprising converting the trace method calls and/or the log method calls to a specified one or more trace and/or log message formats, respectively.

33. (Original) The method as in claim 32 wherein at least one of the trace and/or log message formats comprises a human-readable format for displaying on a display console.

34. (Currently Amended) An article of manufacture having program code stored thereon which, when executed by a machine cause the machine to perform the operations of:

defining a class hierarchy comprising a controller class, a tracing sub-class, and a logging sub-class, wherein the controller class is a parent class to the tracing sub-class and the logging sub-class;

creating an instance of the tracing sub-class associated with specified program code regions of an application, the tracing instance to receive and process tracing method calls generated by the application when the specified program code regions are executed; and

creating an instance of the logging sub-class associated with specified categories of a system that includes a network, the logging instance to receive and process logging method calls from components associated with the categories;

filtering the tracing method calls and the logging method calls according to a specified trace message filtering policy and logging message filtering policy, respectively, the filtering preventing the calls from reaching respective trace files and log files.

35. (Original) The article of manufacture as in claim 34 wherein the instance of the tracing sub-class further identifies a tracing severity level associated with the tracing method calls and to process the tracing method calls based on the tracing severity level.

36. (Original) The article of manufacture as in claim 35 wherein the instance of the tracing subclass processes the tracing method calls by comparing the tracing

severity level of the method calls to a tracing severity threshold, wherein if the tracing severity level is equal to or greater than the tracing severity threshold, a trace message is sent to a first output destination.